Lambda-mu-calculus and Bohm's theorem

René David, Walter Py

The Journal of Symbolic Logic
Volume 00, Number 0, XXX 0000

$\lambda\mu$-CALCULUS AND BÖHM'S THEOREM

RENÉ DAVID AND WALTER PY

Abstract. The $\lambda\mu$-calculus is an extension of the λ-calculus that has been introduced by M Parigot to give an algorithmic content to classical proofs. We show that Böhm's theorem fails in this calculus.

§1. Introduction. The $\lambda\mu$-calculus (its typed and untyped versions) has been introduced by M. Parigot in [6]. Its typed version is an extension of the typed λ-calculus intended to give an algorithmic content to classical proofs.

The main computational rules are β (the usual one of the λ-calculus) and μ. This new rule corresponds (cf. [9]), in the typed version, to the elimination of a logical cut related to the classical rule : If $\Gamma, \alpha : \neg A \vdash M : \perp$, then $\Gamma \vdash \mu\alpha.M : A$. Two other rules ρ and θ (that look like, for the μ-variables, the η-rule) also are introduced. In [6], Parigot proved that the (untyped) $\lambda\mu$-calculus with the rules β, μ, ρ and θ satisfies the Church Rosser property. He also proved ([8]) that every typed term is strongly normalizing.

This paper is concerned with Böhm's theorem. This theorem, in the λ-calculus, says that if two normal closed terms are computationally equivalent (i.e., when applied to any sequence of arguments the first one is solvable iff the second one also is solvable), then they are η-equivalent. We thus also have to consider the η-rule. However in the $\lambda\mu$-calculus, the $\beta\eta\mu\rho\theta$-reduction has not the Church-Rosser's property, because of the following critical pair.

$$\begin{array}{ccc} \lambda x.\,(\mu\alpha.M\ x) & \xrightarrow[\mu]{} & \lambda x.\mu\alpha.M[x/^*\alpha] \\ \big\downarrow \eta & & \\ \mu\alpha.M & & \end{array}$$

In order to be able to state an equivalent form of Böhm's theorem we have to restore the confluence. We thus consider another reduction (called the ν-reduction) which is an η-expansion followed by a μ-reduction : $\mu\alpha.M \ \rightarrow_\nu \ \lambda x.\mu\alpha.M[x/^*\alpha]$ stands for $\mu\alpha.M \ \rightarrow_{\eta_{exp}} \ \lambda x.(\mu\alpha.M \ \ x) \ \rightarrow_\mu \ \lambda x.\mu\alpha.M[x/^*\alpha]$. This reduction, which corresponds exactly to the reduction defined by Prawitz ([9] Chap.ĨII, § 1, Theorem I), has also been considered by Parigot ([6]) but only in the typed version. It is proved in [10] that the $\beta\eta\nu\mu\rho\theta$-reduction satisfies the Church-Rosser's property.

In the presence of ν, Parigot's μ-reduction is no longer needed. Indeed, any μ-reduction may be simulated by a ν-reduction immediately followed by a β-reduction.

0022-4812/00/0000-0000/$00.00

Since a ν-reduction can always be made in front of a μ-abstraction, there are no normal terms. However it is also proved in [10] that every $\beta\mu$-normal term can be reduced to a term in a canonical normal form and thus Böhm's theorem could make sense. Also note that, in the simply typed case, the notion of ν-normal form makes sense since the complexity of the type of the μ-variable involved in a ν-reduction decreases after each step : $\mu\alpha^{\neg(a\to b)}.M \to_\nu \lambda x^a.\mu\alpha^{\neg b}.M[x/^*\alpha]$.

The main result of this paper is that Böhm's theorem fails in the $\lambda\mu$-calculus : We can find two closed terms in canonical normal form that are not $\beta\eta\nu\mu\rho\theta$-equivalent but are operationally equivalent, i.e., they cannot be distinguished by any (not only applicative) context.

We also give a very elementary proof of the following fact : two terms are computationally equivalent iff they are operationally equivalent. This result is sometimes called the operational extensionality or the context lemma. Note that the terminology itself (computational equivalence, operational equivalence, ...) sometimes depends on the author. Some references may be found in [1]. Finally note that, in the λ-calculus, this result is an immediate consequence of ... Böhm's theorem and, as far as we know, the only known proof was by using Böhm's theorem.

Warning The proofs of confluence are not reproduced here because they are long and technical (the reader can find them in [10]). However this paper is self contained : The confluence properties are not used in the *proofs* of the main results. They are only used to give a *sense* to these results. In section 3 we thus only recall these properties and give the main problems occurring in their proofs. In section 4 we prove the failure of Böhm's theorem and the equivalence between the computational and the operational equivalence.

§2. The $\lambda\mu$-calculus.

The set T of $\lambda\mu$-terms is given by the following grammar :

$$T = x \mid \lambda x.\, T \mid (T\ T) \mid \mu\alpha.[\beta]\, T$$

where x ranges over a set V_λ of λ-variables and α, β range over a set V_μ of μ-variables (disjoint from V_λ). Note that in [6], [10] the application was denoted as in [4] by $(u)v$. We adopt here the usual notation $(u\ v)$.

The reduction rules are the following.

$$\begin{aligned}
(\lambda x.M\ \ N) &\to_\beta\ M[x := N]\\
\lambda x.(M\ x) &\to_\eta\ M\ \text{(if } x \text{ is not free in } M)\\
(\mu\alpha.M\ N) &\to_\mu\ \mu\alpha.M[N/^*\alpha]\\
\mu\alpha.M &\to_\nu\ \lambda x.\mu\alpha.M[x/^*\alpha]\\
[\beta]\mu\alpha.M &\to_\rho\ M[\beta/\alpha]\\
\mu\alpha.[\alpha]M &\to_\theta\ M\ \text{(if } \alpha \text{ is not free in } M)
\end{aligned}$$

The substitutions are defined in the following table (where σ is $x := N$, τ is $N/^*\alpha$, ρ is β/α and $\delta \neq \alpha$). Note that μ is a binding operator and that substitutions are thus done with the usual rules, in particular the renaming of bound variables to avoid the capture of free variables.

M	$M[\sigma]$	$M[\tau]$	$M[\rho]$
x	N	x	x
y	y	y	y
$\lambda y.O$	$\lambda y.O[\sigma]$	$\lambda y.O[\tau]$	$\lambda y.O[\rho]$
$(O\ P)$	$(O[\sigma]\ P[\sigma])$	$(O[\tau]\ P[\tau])$	$(O[\rho]\ P[\rho])$
$\mu\gamma.O$	$\mu\gamma.O[\sigma]$	$\mu\gamma.O[\tau]$	$\mu\gamma.O[\rho]$
$[\alpha]O$	$[\alpha]O[\sigma]$	$[\alpha](O[\tau]\ N)$	$[\beta]O[\rho]$
$[\delta]O$	$[\delta]O[\sigma]$	$[\delta]O[\tau]$	$[\delta]O[\rho]$

§3. **Confluence properties of the $\lambda\mu$-calculus.** In this section we give the main confluence results of the calculus. As mentioned before the proofs can be found in [10]. It is important to note that, in the ρ-reduction, β is a free variable and thus $\lambda\mu$ is not a combinatory reduction system in the sense of Klop ([3]). Thus, standard methods cannot be used.

NOTATION 3.1. *$(M \to_{\beta\mu\nu\eta\rho\theta} N)$ (respectively $M \to^*_{\beta\mu\nu\eta\rho\theta} N$) means that M reduces to N by one step (respectively some steps, possibly 0) of either a β or μ or ν or η or ρ or θ reduction. The reduction $\to^*_{\beta\mu\nu\eta\rho\theta}$ is also called the $\beta\mu\nu\eta\rho\theta$-reduction. Similarly, for example, for $M \to^*_{\beta\mu} N$.*

THEOREM 3.2. *The $\beta\mu\rho\theta$-reduction satisfies the Church-Rosser property.*

The proof given in [6] was not completely correct. It uses the method of parallel reductions of Tait and Martin-Löf. But this method does not work in this context for the following reason. Denote by $\Longrightarrow$ the parallel reduction. $M \Longrightarrow M'$ does not imply $M[N/^*\alpha] \Longrightarrow M'[N/^*\alpha]$. For example $M = [\alpha]\mu\beta.O \Longrightarrow_\rho O[\alpha/\beta] = M'$ but $M[N/^*\alpha]$ reduces to $M'[N/^*\alpha]$ in two steps :

$$\begin{array}{rcl} M[N/^*\alpha] & = & [\alpha](\mu\beta.O[N/^*\alpha]\ N\ ) \\ & \Longrightarrow_\mu & [\alpha]\mu\beta.O[N/^*\alpha][N/^*\beta] \\ & \Longrightarrow_\rho & O[N/^*\alpha][N/^*\beta][\alpha/\beta] \\ & = & M'[N/^*\alpha] \end{array}$$

The proof given in [10] uses an extension of the method of Tait and Martin-Löf due to Aczel (see [3] or [2]).

THEOREM 3.3. *The $\beta\eta\mu\nu\rho\theta$-reduction satisfies the Church-Rosser property over μ-closed terms.*

Note that this result is false for open terms. For example the following diagram (where $I = \lambda y.\ y$) cannot be closed because the μ-variable β is free.

$$\begin{array}{ccccc} [\beta]\ \lambda x.\ (\mu\alpha.I\ x) & \xrightarrow[\eta]{} & [\beta]\ \mu\alpha.\ I & \xrightarrow[\rho]{} & I \\ \downarrow{\scriptstyle\mu} & & & & \\ [\beta]\ \lambda x.\mu\alpha.I & & & & \end{array}$$

Even the extended parallel-reduction method mentioned before does not work because the following critical pair occurs between ν and ρ.

$$\begin{array}{ccc}
[\beta]\mu\alpha.M & \xrightarrow[\rho]{} & M[\beta/\alpha] \\
\big\downarrow{\scriptstyle \nu} & & \\
[\beta]\lambda x.\mu\alpha.M[x/^*\alpha] & &
\end{array}$$

The ρ-redex $[\beta]\mu\alpha.M$ is hidden in $[\beta]\lambda x.\mu\alpha.M[x/^*\alpha]$. It is possible to close the diagram only if β is a bounded variable. For example, in the following diagram,

$$\begin{array}{ccc}
\mu\beta.[\beta]\mu\alpha.M & \xrightarrow[\rho]{} & \mu\beta.M[\beta/\alpha] \\
\big\downarrow{\scriptstyle \nu} & & \big\downarrow{\scriptstyle \nu} \\
\mu\beta.[\beta]\lambda x.\mu\alpha.M[x/^*\alpha] & \xrightarrow[\nu]{} \circ \xrightarrow[\beta]{} \circ \xrightarrow[\rho]{} & \lambda y.\mu\beta.M[\beta/\alpha][y/^*\beta]
\end{array}$$

we close by the reduction :

$$\begin{array}{lll}
\mu\beta.[\beta]\lambda x.\mu\alpha.M[x/^*\alpha] & \to_\nu & \lambda y.\mu\beta.[\beta](\lambda x.\mu\alpha.M[x/^*\alpha][y/^*\beta]\ y) \\
 & \to_\beta & \lambda y.\mu\beta.[\beta]\mu\alpha.M[x/^*\alpha][y/^*\beta][x := y] \\
 & \to_\rho & \lambda y.\mu\beta.M[x/^*\alpha][y/^*\beta][x := y][\beta/\alpha] \\
 & = & \lambda y.\mu\beta.M[\beta/\alpha][y/^*\beta]
\end{array}$$

The main problems (that make the proof difficult) are the following :

- a ρ-redex $[\beta]\mu\alpha$ may be hidden by many λ.
- the ν-reductions that are necessary to make this redex "visible" are non local, since they are done in front of the $\mu\beta$ and this may occur far from the ρ-redex.
- to close a critical pair between ν and ρ a β-reduction is necessary and thus the reduction rules cannot be separated.

NOTATION 3.4. *$\vec{\lambda}$ (respectively $\overrightarrow{\lambda\mu}$) represents an arbitrary sequence of λ abstraction (respectively of λ abstractions and prefixes as $\mu\alpha.[\beta]$)*

DEFINITION 3.5. *A $\lambda\mu$-term is in canonical normal form if* :

1. *M is $\beta\eta\mu\rho\theta$-normal.*
2. *$M = \vec{\lambda}.(y\ N_1...N_k)$ or $M = \vec{\lambda}\ \mu\gamma.[\beta]\ (y\ N_1...N_k)$.*
3. *The N_i are in canonical normal form.*

The point 2 in this definition means that M begins with some λ and, at most one, $\mu\alpha[\beta]$.

THEOREM 3.6. *Let M be a μ-closed $\beta\mu$-normal term. There is a term N such that $M \to^*_{\beta\mu\nu\eta\rho\theta} N$ and N is in canonical normal form.*

Note that again this result is only true for μ-closed terms. Also note that the term N is not unique. For example : $\lambda x.\ \mu\alpha.\ [\alpha]\ (x\ \mu\beta.[\alpha]\ x)$ and $\lambda x\lambda y.\ \mu\alpha.[\alpha]\ (x\ \mu\beta.[\alpha]\ (x\ y)\ y)$ both are in canonical normal form but are ν-equivalent. See section 5 about this problem.

DEFINITION 3.7. *M is $\beta\eta\mu\nu\rho\theta$-solvable (respectively $\beta\mu$-solvable) if $M \to^*_{\beta\eta\mu\nu\rho\theta} M'$ (respectively $M \to^*_{\beta\mu} M'$) where M' is in head $\beta\mu$-normal form, i.e., $M' = \overrightarrow{\lambda\mu}\ (x\ \vec{N})$ for some sequence $\vec{N}$.*

Note that the prefix $\overrightarrow{\lambda\mu}$ in a head $\beta\mu$-normal form may contain θ and ρ-redexes.

The following theorems show that, somehow, the main rules of the calculus are β and μ.

THEOREM 3.8. *η, ρ and θ can be postponed, i.e., if $M \rightarrow^*_{\beta\mu\nu\eta\rho\theta} M'$ there are terms N, P, Q such that $M \rightarrow^*_{\beta\mu\nu} N \rightarrow_\eta P \rightarrow^*_\rho Q \rightarrow^*_\theta M'$.*

THEOREM 3.9. *A term t is $\beta\mu$-solvable iff it is $\beta\eta\mu\nu\rho\theta$-solvable.*

DEFINITION 3.10. 1. *Every term can be uniquely written as $\overrightarrow{\lambda\mu}\ (R\ \overrightarrow{u})$ where R is either a $\beta\mu$-redex (called the head redex) or a variable (called the head variable).*
2. *The head reduction consists in reducing, at each step, the head redex. $M \succ M'$ (respectively $M \succ^+ M'$) means that M reduces to M' by some steps (possibly 0) (respectively at least one step) of head reduction.*

The following lemma (which is an easy extension of the same result in the λ-calculus) is often used in the next section.

LEMMA 3.11. 1. *M is solvable iff the head reduction of M terminates in a term in head $\beta\mu$-normal form.*
2. *If $M \succ M'$ then $M[x := N] \succ M'[x := N]$.*
3. *If $M[x := N]$ is solvable, then M is solvable.*

§4. Böhm's theorem fails in the $\lambda\mu$-calculus.

DEFINITION 4.1. 1. *A context is a term with some holes (a hole is denoted by []). An applicative context is a context of the form ([] $N_1 ... N_k$) where the N_i have no holes.*
2. *Two closed terms M and M' are operationally equivalent (respectively computationally equivalent) if for every closed context (respectively applicative closed context) C, $C[M]$ is solvable iff $C[M']$ is solvable. This will be denoted by $M \sim M'$ (respectively $M \sim_a M'$)*

Note that, by theorem 3.9 and lemma 3.11, $M \sim M'$ iff ($C[M]$ reduces by head reduction to a term in head $\beta\mu$-normal form iff $C[M']$ reduces by head reduction to a term in head $\beta\mu$-normal form.

Theorem 4.4 below, which is the main result of this paper, shows that Böhm's theorem fails in the $\lambda\mu$-calculus. Note that it is easy to check that the term W is typeable (in the extension to classical logic of $\lambda_\rightarrow$) and thus the cause of failure is not the untypability. We first prove that the computational equivalence is included in the operational one.

We first need the following definition.

DEFINITION 4.2. *Let C be a context and M be a closed term such that $C[M]$ is solvable. Denotes by $\Phi_M(C)$ the number of times M comes in head position during the head reduction of $C[M]$. More precisely, $\Phi_M(C)$ is defined by induction on the length of the head reduction of $C[M]$ as follows :*

- *If the head variable of C is not [] or if $C \succ \overrightarrow{\lambda\mu}$ [], then $\Phi_M(C) = 0$.*
- *Otherwise $C \succ \overrightarrow{\lambda\mu}$ ([] $\overrightarrow{A}$) for some non-empty sequence $\overrightarrow{A}$. Then $\Phi_M(C) = 1 + \Phi_M(D)$ where D is the head $\beta\mu$-normal form of $(M\ \overrightarrow{A})$.*

Remark Note that Φ_M is well defined since, by the lemma 3.11, C is solvable. Moreover, since M is closed, its head $\beta\mu$-normal form begins with a λ or a μ and thus $C[M] \succ^+ \overrightarrow{\lambda\mu}\ D[M]$ and the reduction of $D[M]$ is shorter than the one of $C[M]$.

THEOREM 4.3. *Let M and M' be closed $\lambda\mu$-terms. If $M \sim_a M'$ then $M \sim M'$.*

PROOF. We prove that if $C[M]$ is solvable, then so is $C[M']$ by induction on $\Phi_M(C)$.

- If the head variable of C is not $[]$ or if $C \succ \overrightarrow{\lambda\mu}[]$, the result follows immediately from lemma 3.11 .
- Otherwise : $C \succ \overrightarrow{\lambda\mu}([]\overrightarrow{A})$. Let D be the head $\beta\mu$-normal form of $(M\overrightarrow{A})$. Then $D[M]$ is solvable and $\Phi_M(D) < \Phi_M(C)$. By the induction hypothesis $D[M']$ is solvable and thus $(M\overrightarrow{A}[M'])$ is solvable. Since $M \sim_a M'$, $(M'\overrightarrow{A}[M'])$ also is solvable and, since $C[M'] \succ \overrightarrow{\lambda\mu}(M'\overrightarrow{A}[M'])$, we are done.

⊣

THEOREM 4.4. *Let $0 = \lambda a\lambda b.b$, $1 = \lambda a\lambda b.a$ and $U_0 = \mu\delta.[\alpha]0$. Let $W = \lambda x.\mu\alpha.[\alpha](x\mu\beta.[\alpha](xU_0y)U_0)$, $W_0 = W[y := 0]$ and $W_1 = W[y := 1]$. Then*

1. *W_0 and W_1 are closed terms in canonical normal form.*
2. *W_0 and W_1 are not $\beta\mu\nu\eta\rho\theta$-equivalent.*
3. *$W_0 \sim W_1$.*

PROOF. 1. and 2. are trivial. By theorem 4.3 it is enough to show that if A is a term and $\overrightarrow{B}$ is a sequence of terms such that $(W_0\ A\ \overrightarrow{B})$ is solvable, then $(W_1\ A\ \overrightarrow{B})$ also is solvable. Let $C = (W\ A\ \overrightarrow{B})$. Since $(W_0\ A\ \overrightarrow{B}) = C[y := 0]$ it follows from lemma 3.11 that C is solvable. It is enough to show that the head variable of C cannot be y.

$C \succ \mu\alpha.[\alpha]\ (A\ \ Z\ U\ \overrightarrow{B})$ where $Z = \mu\beta.[\alpha]\ (A\ U\ y\ \overrightarrow{B})$ and $U = \mu\delta.[\alpha]\ (0\ \overrightarrow{B})$.

Let $C_1 = (A\ z\ u\ \overrightarrow{B})$ where z and u are fresh variables. Letσ be the substitution $[z := Z, u := U]$ and τ be the substitution $[z := U, u := y]$. Note that $Z = \mu\beta.[\alpha]\ C_1[\tau]$ and that the free λ-variables of C_1 are z and u.

Since $C \succ \mu\alpha.[\alpha]\ C_1[\sigma]$, by lemma 3.11, C_1 is solvable.

1. If the head variable of C_1 is bounded the result is clear.
2. If the head variable of C_1 is u. Then, for some $\overrightarrow{\lambda\mu}$ and some sequence $\vec{D}$ of arguments $C \succ \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ (U\ \vec{D}[\sigma])$. Since $U = \mu\delta.[\alpha]\ (0\ \overrightarrow{B})$ and δ does not appear in $(0\ \overrightarrow{B})$, $C \succ \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ U$. Since y does not appear in U, the result follows.
3. If the head variable of C_1 is z. Then, for some $\overrightarrow{\lambda\mu}$ and some sequence $\vec{D}$ of arguments $C \succ \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ (Z\ \vec{D}[\sigma])$. Since $Z = \mu\beta.[\alpha]\ (A\ U\ y\ \overrightarrow{B})$ and β does not appear in $(A\ U\ y\ \overrightarrow{B})$, $C \succ \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ Z = \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ \mu\beta.[\alpha]\ C_1[\tau]$. Thus $C \succ\ \overrightarrow{K}\ (z\ \vec{D})[\tau] = \overrightarrow{K}\ (U\ \vec{D}[\tau])$ where $\overrightarrow{K} = \mu\alpha.[\alpha]\ \overrightarrow{\lambda\mu}\ \mu\beta.[\alpha]\ \overrightarrow{\lambda\mu_1}$ and $\overrightarrow{\lambda\mu_1}$ is a renaming (to avoid capture) of $\overrightarrow{\lambda\mu}$. The result follows then in the same way as in the previous case.

⊣

§5. Conclusion. The intuitive meaning of Böhm's theorem in the λ-calculus is that, by giving to a term the appropriate arguments, we may put any node of its Böhm tree in head position. The first arguments are used to go to the first node of the path, the next arguments are used to go to the next node and so on ... to the specified node. The term W given in section 4 shows that we cannot do the same thing in the $\lambda\mu$-calculus : We showed that the variable y does not come in head position. The main reason is that, in a term as $(\mu\alpha\ M\ \vec{N})$, the whole stack $\vec{N}$ of arguments is given to the sub-terms $[\alpha]P$ of M.

It is also important to note that, in the λ-calculus, the arguments depend only on the path to the specified node and not on the other nodes. In our counter-example for the $\lambda\mu$-calculus, we have not only use the path to y but also the other nodes in W since we extensively used the fact that $(\mu\beta.M\ \overrightarrow{A}) \succ \mu\beta.M$ when β is not free in M.

Some open questions.

1. Is it possible to recover Böhm's theorem by giving other reduction rules? By defining an equivalence relation (that does not necessarily come from other reduction rules) on the $\lambda\mu$-terms in canonical normal forms?
2. The canonical normal form is not unique but distinct forms of the same term look like η-equivalent. Is it possible to get the unicity by giving other reduction rules?
3. It seems that the terms $\mu\alpha.M$ where $[\alpha]$ does not occur in M play an important role. Is it possible to identify all these terms to get such relations? Can we learn something from semantics? In a semantic model where $\perp$ would be interpreted as the initial object of some category this would be the case.

In [5], K. Nour introduces new rules to the $\lambda\mu$-calculus and he gets a calculus satisfying the following properties : Strong normalization, subject reduction and existence of a *parallel or* in the typed calculus. The calculus does not satisfy the Church-Rosser property but in the usual data types (integers, booleans, lists of integers, ...) the unicity of representation is preserved.

This calculus could be a good candidate to answer the previous questions since, in this calculus, the term W given in section 4 reduces to $\lambda x.0$, but the meaning of Böhm's theorem in a calculus that is not confluent is unclear.

REFERENCES

[1] S. Abramsky and L. Ong, *Full abstraction in the lazy lambda calculus*, ***Information and Computation***, vol. 105 (1993), no. 2.

[2] P. Aczel, *A general church-rosser theorem*, ***Technical report***, University of Manchester, 1978.

[3] J. W. Klop, V. van Oostrom, and F. van Raamsdonk, *Combinatory reduction systems, introduction and survey*, ***Theoretical Computer Science***, vol. 121 (1993).

[4] J. L. Krivine, ***Lambda-calcul, types et modèles***, Masson, Paris, 1990.

[5] K. Nour, *Non deterministic classical logic : The $\lambda\partial$-calculus*, Private communication.

[6] M. Parigot, ***$\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction***, Lecture Notes in Artificial Intelligence, no. 624, Springer-Verlag, 1992.

[7] ———, ***Classical proofs as programs***, Lecture Notes in Computer Science, no. 713, Springer Verlag, 1993.

[8] ———, *Proofs of strong normalization for second order classical natural deduction*, this Journal, vol. 62 (1997), no. 4.

[9] D. Prawitz, ***Natural deduction, a proof-theoritical study***, Almqvist & Wiksell, Stockholm, 1965.

[10] W. Py, *Confluence en $\lambda\mu$-calcul*, ***Ph.D. thesis***, 1998.

ADDRESS NEEDED

ADDRESS NEEDED